

ABSTRACT

Critical sections of multi-threaded programs, normally protected by locks providing access by only one thread, are speculatively executed concurrently by multiple threads with elision of the lock acquisition and release. Upon a completion of the speculative execution without actual conflict as may be identified using standard cache protocols, the speculative execution is committed, otherwise the speculative execution is squashed. Speculative execution with elision of the lock acquisition, allows a greater degree of parallel execution in multi-threaded programs with aggressive lock usage.

5136566_1.DOC